**Приложение Г. Скрипт базы данных**

АННОТАЦИЯ

В данном программном документе приведен скрипт базы данных программы склада микроэлектроники.

В разделе «Скрипт базы данных» указывают название скрипта, краткую характеристику области применения скрипта и сам скрипт базы данных.

СКРИПТ БАЗЫ ДАННЫХ

* 1. Наименование скрипта

Наименование – FurnitureProductionBD.sql

* 1. Область применения скрипта

Скрипт должен эксплуатироваться в базе данных программы склада микроэлектроники и API.

* 1. Скрипт базы данных

create database MicroElectronsDB;

use MicroElectronsDB;

create table Post(

id int primary key auto\_increment,

name nvarchar(100) not null

);

create table EmployeeStatus(

id int primary key auto\_increment,

name nvarchar(100) not null

);

create table Employee(

id int primary key auto\_increment,

lastName nvarchar(50) not null,

firstName nvarchar(50) not null,

patronymic nvarchar(50) null,

birthday date not null,

postId int not null,

foreign key (postId) references Post (id),

statusId int not null,

foreign key (statusId) references EmployeeStatus (id)

);

create table User(

id int primary key auto\_increment,

login varchar(50) not null unique,

password varchar(64) not null,

employeeId int not null,

foreign key (employeeId) references Employee (id)

);

create table Labor(

id int primary key auto\_increment,

salary double not null,

dateConfirm date not null,

employeeId int not null,

foreign key (employeeId) references Employee (id)

);

create table Dismissal(

id int primary key auto\_increment,

dateConfirm date not null,

employeeId int not null,

foreign key (employeeId) references Employee (id)

);

create table Holiday(

id int primary key auto\_increment,

dateStart date not null,

dateEnd date not null,

employeeId int not null,

foreign key (employeeId) references Employee (id)

);

create table Visitor(

id int primary key auto\_increment,

lastName nvarchar(50) not null,

firstName nvarchar(50) not null,

patronymic nvarchar(50) null,

passport varchar(10) not null

);

create table VisitorJournal(

id int primary key auto\_increment,

dateTimeEntry datetime not null,

dateTimeExit datetime null,

employeeEntryId int not null,

foreign key (employeeEntryId) references Employee (id),

employeeExitId int null,

foreign key (employeeExitId) references Employee (id),

visitorId int not null,

foreign key (visitorId) references Visitor (id)

);

create table ProductCategory(

id int primary key auto\_increment,

name nvarchar(100) not null

);

create table Product(

id int primary key auto\_increment,

name nvarchar(100) not null,

categoryId int not null,

foreign key (categoryId) references ProductCategory (id)

);

create table StorageMethod(

id int primary key auto\_increment,

name nvarchar(100) not null

);

create table Warehouse(

id int primary key auto\_increment,

quantity int default 0,

productId int not null,

foreign key (productId) references Product (id),

storageMethodId int not null,

foreign key (storageMethodId) references StorageMethod (id)

);

create table TaskStatus(

id int primary key auto\_increment,

name nvarchar(100) not null

);

create table Task(

id int primary key auto\_increment,

dateStart date not null,

dateEnd date null,

dateDeadline date not null,

quantity int not null,

warehouseId int not null,

foreign key (warehouseId) references Warehouse (id),

employeeId int not null,

foreign key (employeeId) references Employee (id),

statusId int not null,

foreign key (statusId) references TaskStatus (id)

);

create table Counterparty(

id int primary key auto\_increment,

name nvarchar(100) not null,

tin varchar(10) not null,

address nvarchar(200) not null,

bic varchar(9)

);

create table Supply(

id int primary key auto\_increment,

isSell bool default 0,

dateSupply date not null,

counterpartyId int not null,

foreign key (counterpartyId) references Counterparty (id)

);

create table SupplyCompos(

id int primary key auto\_increment,

quantity int not null,

summa double not null,

productId int not null,

foreign key (productId) references Product (id),

supplyId int not null,

foreign key (supplyId) references Supply (id)

);

create table OperationType(

id int primary key auto\_increment,

name nvarchar(100) not null

);

create table ComeJournal(

id int primary key auto\_increment,

subjectName nvarchar(100) not null,

quantity int not null,

dateTimeConfirm datetime not null,

isCome bool default 1,

operationId int not null,

foreign key (operationId) references OperationType (id)

);

drop trigger if exists insert\_supply;

create trigger insert\_supply after insert

on SupplyCompos for each row begin

insert into ComeJournal(subjectName, quantity, dateTimeConfirm, isCome, operationId) values

((select name from Product where Product.id=NEW.productId),

(NEW.quantity),

(select dateSupply from Supply where Supply.id=NEW.supplyId),

not(select isSell from Supply where Supply.id=NEW.supplyId),

(select id from OperationType where name='Поставки'));

update Warehouse set quantity = quantity + if((select isSell from Supply where Supply.id=NEW.supplyId), -NEW.quantity, NEW.quantity)

where productId=NEW.productId;

end;

drop trigger if exists update\_task;

create trigger update\_task after update

on Task for each row begin

insert into ComeJournal(subjectName, quantity, dateTimeConfirm, isCome, operationId) values

((select name from Product where Product.id=(select productId from Warehouse where Warehouse.id=NEW.warehouseId)),

(NEW.quantity),

(select dateEnd from Task where Task.id=NEW.id),

1,

(select id from OperationType where name='Производство'));

update Warehouse set quantity = quantity + NEW.quantity

where Warehouse.id=NEW.warehouseId;

end;

create trigger insert\_product after insert

on Product for each row begin

insert into Warehouse(productId, quantity, storageMethodId) values

(NEW.id, 0, 1);

end;